APSC 143 – Introduction to Programming for Engineers

Assignment 8 – Functions

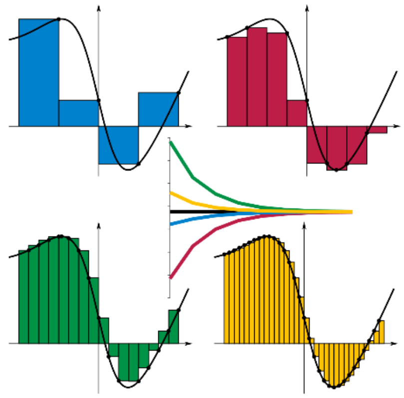

Figure 1: Convergence of Riemann sum for all sample position choices as intervals shrink, KSmrq (4 July 2007)

Background Theory:

$$\int_a^b f(x)dx = \lim_{\Delta x \to 0} \sum_{i=1}^{n} f(x_i) * \Delta x \text{ where } \Delta x = \frac{b-a}{n}, \text{and } x_i = a + \Delta x * i$$

The Riemann Sum of a function is the sum of rectangles with a defined width and a height of the functions value at that point. The more rectangles/intervals in the Riemann Sum, the better the approximation for the area underneath a curve in the xy-plane.

The integral of a function can be defined the sum of rectangles with an infinitesimally small width and a height of the functions value at that point. In other words, an integral is a Riemann sum where the width of the rectangles is infinitesimally small, and the number of intervals approaches infinity. The Integral of a function can be found by taking the anti-derivative of all terms in the integral, then subtracting the result evaluated at the upper bound by the result evaluated at the lower bound.

Unfortunately, there are some functions that are not integrable using the integral definition above. There are two ways to get around this to obtain an approximation for the integral.

- Use the Riemann Sum Definition with a definite rectangle width
- Approximate the function using a series approximation, and integrating that

One of these series approximations is called the Taylor Series, it is a series of infinite terms that approximates a function more precisely the more terms in the series are included. The Taylor Series of common functions are known, and other functions can be found by manipulating the summation definition like below. You will learn more about the Taylor Series in APSC 172 this winter.

$$e^x = \sum_{n=0}^{\infty} \frac{x^n}{n!} \quad \to \quad e^{x^2} = \sum_{n=0}^{\infty} \frac{(x^2)^n}{n!}$$

You will examine the most notable non-integrable function, $f(x) = e^{x^2}$, and use the Riemann Sum definition of an integral to evaluate the area underneath this curve. Then, you will approximate the function using a Taylor Series, evaluate its Riemann Sum, and compare the results.

APSC 143 – Introduction to Programming for Engineers

Instructions:

In this assignment, you will create a program that includes a function that evaluates the Riemann Sum of $f(x) = e^{x^2}$ between two points with a certain number of intervals, as well as any sub-functions necessary. You will then use a Taylor Series Approximation to find an alternate representation for the given function, compute its Riemann Sum, and compare the two results.

Below is a guideline for how to approach this problem, though you may choose to approach it however you wish. Make sure to express your results using the format specified below:

- In the main function
  - Ask the user to input the bounds at which to evaluate the Riemann Sum.
    - Assume that the bounds entered are valid ($1^{st} < 2^{nd}$)
  - Call the Riemann Sum function to compute the Riemann Sums with 5, 25, and 100 intervals at the specified bounds.
  - Call the Taylor Series function to compute the Taylor Series approximation at the **LOWER BOUND** inputted by the user, with 3, 5, and 10 terms in the series.
  - Call the Taylor Series Riemann Sum function to compute the Riemann Sums with 5, 25, and 100 intervals at the specified bounds. Use 10 terms in the Taylor Series.
  - Compare the two Riemann Sums with the same number of intervals and calculate the error between the two Riemann Sums.
- Create a function that passes in the two bounds, as well as the number of intervals to compute the **LEFT** Riemann Sum of the **Actual Function Representation**
  - Use the three parameters to determine the width of the rectangles in the Riemann Sum
  - Left Riemann Sum rectangles have a height equal to the function on the Left side of the rectangle
- Create a function to compute the Taylor Series approximation of the given function at a given value of x
- Create a function that passes in the two bounds, as well as the number of intervals to compute the **LEFT** Riemann sum of the **Taylor Series Approximation**.
  - This should be very similar to the other Riemann Sum function, just using the Taylor Series Approximation function as a nested function

APSC 143 – Introduction to Programming for Engineers

Comments are mandatory for this assignment. Add comments as necessary for key pieces in your code, such as variable declaration, conditional statements, and looping conditions to explain what the program is doing.

Your output must match the sample output below exactly; otherwise, the auto grading software will not be able to grade your assignment, which may affect your mark.

Example Output:

```
Enter the bounds to compute the Riemann Sum: -2 2

Riemann Sum with 5 intervals: 52.31

Riemann Sum with 25 intervals: 33.83

Riemann Sum with 100 intervals: 32.96

--------------------------------------------------------

Taylor Series Approximation at x = -2.00 with 3 terms: 13.00

Taylor Series Approximation at x = -2.00 with 5 terms: 34.33

Taylor Series Approximation at x = -2.00 with 10 terms: 54.15

--------------------------------------------------------

Taylor Series Riemann Sum with 5 intervals: 51.95

Taylor Series Riemann Sum with 25 intervals: 33.73

Taylor Series Riemann Sum with 100 intervals: 32.88

--------------------------------------------------------

Error in Riemann Sums with 5 intervals: -0.68%

Error in Riemann Sums with 25 intervals: -0.30%

Error in Riemann Sums with 100 intervals: -0.25%
```

Submission Instructions:

Create your program using CLion and upload it to Gradescope for grading. **Your program file must be named** "**apsc143assign8.c**" in order for your assignment to be graded. Do not include any personal information (student number, name, etc.) in your submission. Also, please include a comment in your code attesting to the originality of your work.

Refer to the assignment rubric on OnQ for a detailed breakdown of the grading criteria. Your submission must adhere to the assignment rules as outlined in the submission policy document for this course, which can also be found on OnQ. There is zero tolerance for plagiarism in this course. This auto grading software will automatically flag potential cases of plagiarism, which will be reviewed by the instructors.

More information on assignment submissions can be Found in Week 2, and information on the specific definition and repercussions of plagiarism can be found in the "Begin Here (About This Course)" module

Additional Hints:

- The relation between summation notation and for loops can be defined as:

  `For (int i = 0; i < N; i++) {term}` $\rightarrow \sum_{i=0}^{N} term$

  In a Riemann Sum, you will need to increment the loop parameter by the rectangle width

- These functions are optional, but make the previous steps much more convenient
  - A function that evaluates the given function at a parameter x and returns the result. A factorial function that returns the factorial value of a given integer parameter
  - A function to calculate the error between two values, the second value being the **theoretically correct value**